ATTORNEY'S DOCKET NO: 200309970-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Paul Mantey et al.
Serial No:	10/662,034
Filed:	September 12, 2006
For:	Communications Bus Transceiver
Examiner:	Matthew D. Spittle
Art Unit:	2111

RULE 131 DECLARATION

I, David R. Maciorowski, do hereby declare that:

1. I am David R. Maciorowski. I am an inventor of the subject matter claimed in the above-referenced patent application. I am, and at all relevant times have been, an employee of Hewlett Packard Company, to which I have assigned all right, title, and interest in the subject matter of the above-referenced patent application.

2. Before September 9, 2003, I and the other named inventors of the above-referenced patent application actually reduced to practice, in the United States, a system which implemented all of the limitations of the claims of the above-referenced patent application, as those claims were presented on June 21, 2007.

3. More specifically, before September 9, 2003, I and the other inventors of the above-referenced patent application reduced to practice the Manageability Communications Bus shipped in the 2nd generation Hewlett Packard Keystone and Matterhorn systems, known publicly as the rx8620, rx7620, rp8420, and rp7420 systems. The Manageability Communications Bus, embodiments of which are described in this patent application, improved the efficiency of the I^2C communication protocol, allowing greater bandwidth of communication between cell controllers and the system controller, and improving firmware update times within the system. This system included at least the features described in the attached signed and witnessed Invention Disclosure document.

4. Certain dates have been redacted from the attached Invention Disclosure document. All of the redacted dates are earlier than September 9, 2003.

I further declare under penalty of perjury pursuant to the laws of the United States of America that the foregoing is true and correct, and that this declaration was executed by me on _____, 20____, in the city of _____, state of _____.

_____

David R. Maciorowski (Declarant)

FSTL

Write in Dark Ink on Front Side Only, Please

INVENTION DISCLOSURE

PDNO 20030970 DATE RCVD [redacted] PAGE ONE OF ____ ATTORNEY LPG

*Instructions: The information contained in this document is **COMPANY CONFIDENTIAL** and may not be disclosed to others without prior authorization. Submit this disclosure to the HP Legal Department as soon as possible. No patent protection is possible until a patent application is authorized, prepared, and submitted to the Government.*

Descriptive Title of Invention: Method and Apparatus for an I2C Bus Transceiver with FIFOs, automatic retry, byte timers, fair arbitration, and automatic, programmable CRC generation.

Name of Project: ROME-X

Product Name or Number: RP8420 (Follow on to RP8400)

Was a description of the invention published, or are you planning to publish? If so, the date(s) and publication(s).
No

Was a product including the invention announced, offered for sale, sold, or is such activity proposed? If so, the date(s) and location(s):
No

Was the invention disclosed to anyone outside of HP, or will such disclosure occur? If so, the date(s) and name(s)
No.

If any of the above situations will occur within 3 months, call your IP attorney or the Legal Department now at 1-800-4919 or 970-898-4919.

Was the invention described in a lab book or other record? If so, please identify (lab book #, etc.)

Yes - Verilog files and Word specification documents.

Was the invention built or tested? If so, the date:

Yes - Development is underway

Was this invention made under a government contract? If so, the agency and contract number: No.

Description of Invention: *Please preserve all records of the invention and attach additional pages for the following. Each additional page should be signed and dated by the inventor(s) and witness(es).*

A. Description of the construction and operation of the invention (include appropriate schematic, block, & timing diagrams, drawings, samples, graphs, flowcharts, computer listings, test results, etc.)
B. Advantages of the invention over what has been done before.
C. Problems solved by the invention.
D. Prior solutions and their disadvantages (if available, attach copies of product literature, technical articles, patents, etc.).

Signature of Inventor(s): Pursuant to my (our) employment agreement, I (we) submit this disclosure on this date: [February 11, 2003]

Employee No.	Name	Signature	Telnet	Mailstop	Entity & Lab Name
00312357	Paul J. Mantey		898-3807	MS 3U-B5	40NS-FSTL
00592084	Mike D. Young		898-4766	MS 3U-B5	40NS-FSTL
00244173	David R. Maciorowski		898-6281	MS 3U-B5	40NS-FSTL
Employee No.	Name	Signature	Telnet	Mailstop	Entity & Lab Name
Employee No.	Name	Signature	Telnet	Mailstop	Entity & Lab Name

(If more than four inventors, include additional information on another copy of this form and attach to this document)

Form 3.1 idf.doc, rev. 06/03/00

Write in Dark Ink on Front Side Only, Please

INVENTION DISCLOSURE

COMPANY CONFIDENTIAL PAGE ___ OF ___

Signature of Witness(es): *(Please try to obtain the signature of the person(s) to whom invention was first disclosed.)*

The invention was first explained to, and understood by, me (us) on this date: [redacted]

Full Name	Signature	Date of Signature
Christopher Shawn Krieger		[redacted]
Full Name	Signature	Date of Signature
Michael P. Davis		[redacted]

Inventor & Home Address Information: *(If more than four inventors, include addl. information on a copy of this form & attach to this document)*

Inventor's Full Name

Paul J. Mantey

Street

5212 Greenview Drive

City State Zip

Fort Collins, CO 80525

Do you have a Residential P.O. Address? P.O. BOX City State Zip

No

Greeted as (nickname, middle name, etc.) Citizenship

U.S.

Inventor's Full Name

Michael D. Young

Street

1766 DeWeese Street

City State Zip

Fort Collins, CO 80526

Do you have a Residential P.O. Address? P.O. BOX City State Zip

Greeted as (nickname, middle name, etc.) Citizenship

Mike U.S.

Inventor's Full Name

David R. Macforowski

Street

2043 Blue Mountain Road

City State Zip

Longmont, CO 80504

Do you have a Residential P.O. Address? P.O. BOX City State Zip

Greeted as (nickname, middle name, etc.) Citizenship

Dave U.S.

Inventor's Full Name

Street

City State Zip

Do you have a Residential P.O. Address? P.O. BOX City State Zip

Greeted as (nickname, middle name, etc.) Citizenship

Write in Dark Ink on Front Side Only. Please

Description of Invention: *Please preserve all records of the invention and attach additional pages for the following. Each additional page should be signed and dated by the inventor(s) and witness(es).*

A. Description of the construction and operation of the invention (include appropriate schematic, block, & timing diagrams; drawings; samples; graphs; flowcharts; computer listings; test results; etc.)

B. Advantages of the invention over what has been done before.

C. Problems solved by the invention.

D. Prior solutions and their disadvantages (if available, attach copies of product literature, technical articles, patents, etc.).

Write in Dark Ink on Front Side Only, Please

A) Description of the construction and operation of the invention (include appropriate schematic, block, & timing diagrams; drawings; samples; graphs; flowcharts; computer listings; test results; etc.)

The MCB (Manageability Communications Bus) design consists of two primary (send and receive) state machines, each with an attached FIFO. The state machines and FIFOs interface to a previously purchased TWSI (Two Wire Serial Interface, e.g. I2C) master/slave core as shown in the diagram.

Sending Messages:
To send a message, the host processor writes the destination address. The processor then writes the payload data to the send FIFO. The processor must write a minimum of one byte to the send FIFO before writing the message length registers. The process of writing the message length register initiates the transmission. The send state machine latches the destination address and the first data byte into the master/slave core, and initiates a command to the core to send the data byte to the destination address. The processor continues to load the complete message into the FIFO (the FIFO provides a low-water mark to indicate to the processor that it could accept more data without stalling the transmission bus). Once the message has been completely sent, the send machine initiates a read command to the slave core. The slave core then returns a checksum or CRC byte which has been calculated during the receive process. The sending machine then indicates success (or error) to the processor indicating that the message was successfully transmitted and that the checksum or CRC byte matched the expected value. This interface, from the send state machine to the processor, is implemented via interrupts, interrupt source and error/status registers.

Automatic Retry:
If the master did not successfully negotiate for the bus (arbitration loss) or it targeted a device that was not present, or not operating (slave nak), the master will automatically retry the message at the next instance that it detects the bus is free. There are three conditions for automatic retry: 1) it has to be enabled in the register set, 2) the start of the message must still be in the FIFO, and 3) the number of retries that has occurred for this particular message must not exceed a user defined limit.

Fair Arbitration:
Automatic retry cuts down on the processor overhead required to send messages, but it also results in a highly organized bus. That is, it is quite likely that several different masters will have pending messages in their queue during the time which another device owns the bus. Once the original owner completes the transaction, the other masters will sense the bus going free at the same time, and initiate their transactions at the same time. The bus master that is sending to the lowest address, or is the first to send a unique low data bit, will always win the bus it is possible, therefore, for bus masters to never gain access to the bus due to their priority level. By altering their busfree timer (the timer mechanism that senses when the bus becomes free) in a round-robin fashion (adding a constant multiplied by a priority level) it is possible for these devices to gain access to the bus.

Receiving Messages:
When the slave side of the master/slave TWSI core is targeted as a slave, the receive machine is activated, taking data from the TWSI core and storing it in the receive FIFO. The receive machine will continue responding to the TWSI core, moving data from the core and placing it into the receive FIFO until the end of the message is signaled by the master. If the receive message exceeds the FIFO, a high water mark, and then a full mark both trigger independent interrupts to the processor indicating a full-status in the FIFO. Once the entire message has been sent/received across the bus, the master will command a read of the receive core. The receive core then sends back the checksum or CRC byte back across the bus to the master. This checksum or CRC byte has been continuously updated during the receipt of the message. The specific protocol implemented in this case requires that the checksum or CRC byte always be 0x00. Upon completion of the transaction, the receive message machine will interrupt the processor, indicating that a message is complete, stored in the FIFO, and that the checksum or CRC was 0x00 to indicate a successful message.

Byte Timers:
In the event that a bus master or slave locks up the I2C bus, a byte-timer watch-dog has been implemented to force all of the devices off of the bus and then allow those that are still functional to re-connect to the bus. For instance, if a host processor were to crash before it was able to service a receive interrupt indicating that the receive FIFO was getting full, then the FIFO would continue to fill, and upon filling the FIFO, the receive machine

would stall the I2C bus until the FIFO was emptied sufficiently to allow more data to be written to it. This stall could last indefinitely unless a byte timer was implemented. Byte timers in each core that attached to the bus would trigger at roughly the same time and flush their FIFOs of pending messages. They would also signal to their host processors that a byte timeout event has occurred on the bus. The failed processor would not (likely) see this timeout event. However, the receive FIFO attached to this machine would be flushed, and the transaction would clear itself.

Failed processor detection mechanism:
In the above scenario, detecting which host processor had failed would be difficult. Any subsequent messages written to this device would cause a byte timeout failure to re-occur. Perhaps the other bus transceiver hosts would detect that sends to this particular, failed host, cause repeated byte timeouts and therefore, this device would be removed from the list of acceptable targets. However, there would be no information available as to why the particular device had failed.

However, if the MCB were designed to detect such a failed processor condition, it could automatically do one, or both of the following:

a) Attempt to reset the host processor and bring it back from a crashed state. A signal could be brought out of the MCB machine that when wired to the processors watchdog controller would cause a processor hard reset to occur in the above mentioned scenario. If this failed to bring back the host processor, then the other MCB machine would attempt the second option.
b) Enable a direct connection between the slave I2C device in the master/slave core and the processor bus and send a message to the other bus hosts that a failure condition exists on this particular device. The other devices could then access the failed processor bus via the I2C bus and attempt to diagnose the mechanism that caused the processors' failure. This would greatly benefit the system designers in debugging any system failures.

B) Advantages of the invention over what has been done before
–This invention implements an I2C bus with send and receive FIFOs, multi-master support, automatic retry, and fair arbitration in order to improve both bandwidth and stability of the communications link.

a. The Send and Receive FIFOs reduce the interrupt latency on the bus. They effectively hide the processor interrupt service routine delay through proper setting of high-water (for receive) and low water (for send) marks within the FIFOs.
b. Multi-master support increases the bandwidth of the bus since the devices do not have to request the bus from a single master. Each device can arbitrate for the bus directly and monitor the state of the bus to determine if it has won arbitration.
c. Automatic retry capability allows for a master with a pending transaction to automatically re-arbitrate for the bus once the previous owner of the bus finishes mastering a transaction.
d. The fair arbitration scheme overcomes the inherent I2C protocol problem where the device sending the lowest target address (or if the target addresses are the same, the device which sends the first unique low data bit) wins the bus.
e. Automatic checksum or CRC generation reduces processor overhead in message transactions.

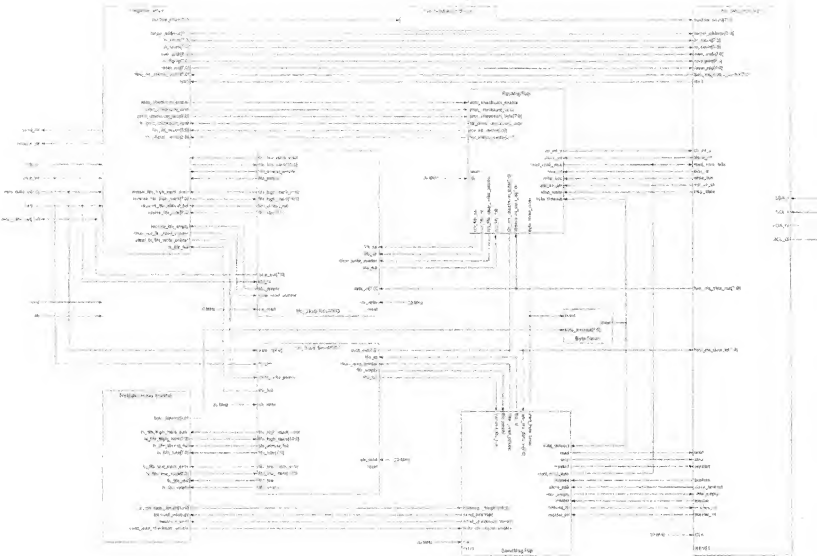
C) Problems solved by the invention.

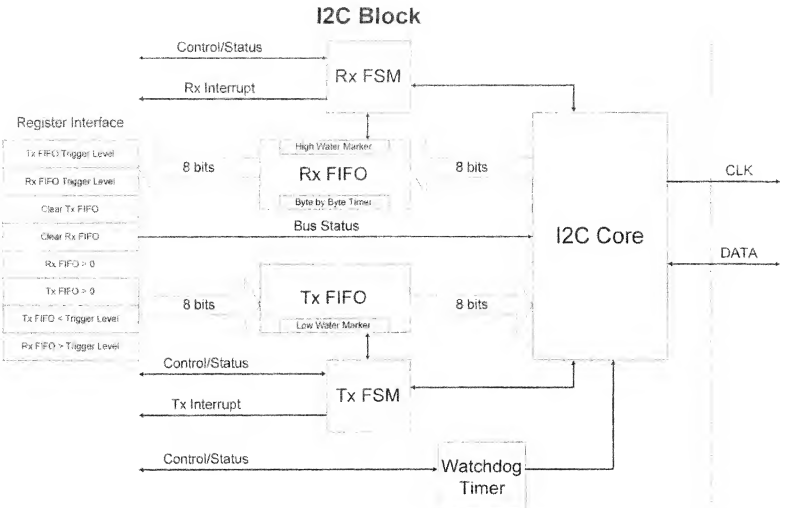
a. This invention allowed for an I2C bus to operate at near peak-bandwidth by eliminating the interrupt-on-byte latency through the addition of FIFOs and by separating the send and receive buffers into separate, state-machine controlled entities.
b. It permitted four masters of a similar type, doing concurrent work (booting a system) to gain access to the bus by providing a fair arbitration.
c. The need for a higher speed link between the manageability processor and the four cell processors was met without modifying the physical layout of the I2C bus between the five devices (increased bus bandwidth without modifying the physical bus topology).

D) Prior solutions and their disadvantages (if available, attach copies of product literature, technical articles, patents, etc.)

a. Older two-wire bus protocols (e.g. SMBus or I2C) supported multi-master configurations (5 masters on one bus), and could operate at 100 or 400 kHz with interrupt-on-byte processor interface. Fair arbitration was guaranteed by the inherent randomness of interrupt service routine latencies within the processor.
    i. Interrupt-on-byte mechanism means that for every byte transmitted, the bus must stall for the length of the longest interrupt service routine time (receiving processor interrupt to read the byte from the receive buffer or transmitting processor interrupt to place the next byte into the transmit buffer). At 100 kHz, the transmission of one byte takes ~100 usec. The typical interrupt service time of a microcontroller in this application can vary from ~75 – ~150 usec (or longer, depending upon load conditions). Thus, the interrupt overhead adds 75 to 150% latency to the bus. In other words, where as a 100 kHz bus has a theoretical peak bandwidth of 11 KB/sec, this interrupt latency reduces the theoretical bandwidth of the bus to approximately 4 - 5.7 KB/sec.
    ii. While the randomness of the delay in the interrupt response time does allow, statistically, for all 5 processors to access the bus in a fair manner, the wait period for the bus to become free is unreasonably long.
b. There are other higher speed protocols available (e.g. Ethernet, USB, etc) with higher bandwidths, but these protocols have generally more expensive IP or routing fabrics (require the use of magnetics, hubs, expensive IP, complicated drivers, etc.)

MCB Block Diagram




I2C Block
Control/Status
Rx FSM
Rx Interrupt
Register Interface
Tx FIFO Trigger Level
Rx FIFO Trigger Level
Clear Tx FIFO
Clear Rx FIFO
Rx FIFO > 0
Tx FIFO > 0
Tx FIFO < Trigger Level
Rx FIFO > Trigger Level
8 bits
High Water Marker
Rx FIFO
Byte by Byte Timer
8 bits
CLK
Bus Status
I2C Core
DATA
8 bits
Tx FIFO
Low Water Marker
8 bits
Control/Status
Tx FSM
Tx Interrupt
Control/Status
Watchdog Timer